

# Epidemic Information Diffusion: A Simple Solution to Support Community-based Recommendations in P2P Overlays

Patrizio Dazzi<sup>1</sup>, Matteo Mordacchini<sup>2</sup>, and Laura Ricci<sup>3</sup>

<sup>1</sup> ISTI-CNR, Pisa, Italy  
`patrizio.dazzi@isti.cnr.it`

<sup>2</sup> IIT-CNR, Pisa, Italy  
`matteo.mordacchini@iit.cnr.it`

<sup>3</sup> University of Pisa  
`matteo.mordacchini@iit.cnr.it`

**Abstract.** Epidemic protocols proved to be very efficient solutions for supporting dynamic and complex information diffusion in highly distributed computing infrastructures, like P2P environments. They are useful bricks for building and maintaining virtual network topologies, in the form of overlay networks as well as to support pervasive diffusion of information when it is injected into the network. This paper proposes a simple architecture exploiting the features of epidemic approaches to foster a collaborative percolation of information between computing nodes belonging to the network aimed at building a system that groups similar users and spread useful information among them.

## 1 Introduction

This paper proposes the recipe for the definition of a simplified system architecture that aims at exploiting a collaborative exchange of information between peers belonging to a highly distributed infrastructure in order to build a computing/network approach able to link similar users in order to foster the process of information percolation within the nodes of the network. More in detail, we push further the idea of realising collaborative recommender mechanisms, by means of solutions fostering interest clustering, that are obtained by means of interactions happening among users. Our approach couples epidemic-based P2P overlay networks to ease the gathering of users with similar interests and then use the connections established to let peer exchange recommendations one each others. Our goal is twofold. On one hand we aim at building an adaptive system supporting the recognition of communities of users' interests in a decentralized, distributed way. The approaches that have been proposed so far in the area of P2P computing (and the epidemic ones, in particular) are able to manage a very large amount of peers and to deal gracefully with churn, whereas centralized systems require expensive and, often, very complex techniques to ensure continuous operation under node and link failures. The service is implemented by means of

the collaborations established between computing nodes, without needing any centralized authority devoted to store all the profiles and the ratings of users as well as to provide centralized-controlled suggestions. On the other hand, our goal is to exploit such communities not limiting our aim to the knowledge sharing about interesting items within them, but also to address some of the traditional problems affecting recommender systems. In particular, the ability to recommend new, almost unknown, items. The system we are sketching, assumes that each neighbor of a computing node (peer)  $P$  pushes recommendations to it focused on the items that might be of potential interest for  $P$ . It is worth to notice that this decision is taken locally, when a neighbor selects or becomes aware from its links of the existence of a new item, whose characteristics, are somehow related with one (or more) of its communities. Then, it can then recommend such item both to  $P$  and to its other neighbors, of all the related communities, as well. This approach would allow a more efficient and rapid percolation of the information within the overlay network. The remain of this paper is organized as follows: in Sec. 2 we shortly present the scientific literature about the subject of this paper; in Sec. 3 we describe the architecture of our proposed system. Finally, in Sec. 4, conclusions are given and potential further exploitations of this work are proposed.

## 2 Related Work

The correlations of interests amongst a group of distributed users has been leveraged in a variety of contexts and for designing or enhancing various distributed systems [1,2]. For peer-to-peer file sharing systems that include file search facilities (e.g., Gnutella, eMule, etc.), an approach to increase recall and precision of the search is to group users based on their past search history or based on their current cache content [3,4]. Another potential use of interest clustering is to form groups of peers that are likely to be interested in the same content in the future, hence forming groups of subscribers in a content-based information diffusion system [5–14]. Moreover, interest correlation can be used to help bootstrapping and self-organization of dissemination structures such as network-delay-aware trees for RSS dissemination [15]. The correlation between the users' past and present accesses has been used for user-centric ranking. In order to improve the customisation of search results, the most probable expectations of users are determined using their search log stored on a centralized server [16,17]. However, the correlation between users is not leveraged to improve the quality of result personalization, hence making the approach viable only for users with sufficiently long search logs. An alternative class of clustering search engines uses semantic information in order to cluster results according to the general domain they belong in (and not as in our approach to cluster users based on their interests). This can be seen as a centralized, user-agnostic approach to improve user experience. The clustering amongst data elements is derived from their vocabulary. It presents the user with results along different interest domains and can help the user to disambiguate these results from a query that may cover several do-

mains, e.g., the query word apple can relate to both food/fruits and computers domains. Examples of such systems are EigenCluster [18], or TermRank [19]. Nonetheless, these systems simply modify the presentation of results so that the user decides herself in which domain the interesting results may fall these results are not in any way automatically tailored to her expectations. They do not also consider the clustering of interest amongst users, but only the clustering in content amongst the data.

Other approaches cluster users on the basis of similarity between their semantics profile. Approaches of this kind of systems includes GridVine [20], the semantic overlay networks [21] and p2pDating [22]. They build a semantic P2P overlay infrastructure that relies on a logical layer storing data.

They make use of heterogeneous but semantically related information sources whereas our approach does not rely on any kind of semantic interpretation. It, in principle, enables a broader exploitation of more heterogeneous data sources. Related with our proposal is Tribler [23], a P2P television recommender system. In contrast with our approach, neighbor lists can be directly filled in by the user herself using an interface. No topology or affinity property is considered. We propose a gossip system that construct and maintain in rest groups of dynamic users based on their past activities, without needing their direct intervention.

### 3 Proposed approach

This section introduces the main pillars that would be needed to support the construction and the exploitation of an overlay network made of peers that share common interests. Figure 1 sketches the architecture of an overlay network organised accordingly. As can be observed in figure, the links between peers are established when they are characterised by a common interest. This information is derived recognising the accesses performed by peers to the same content in the past. The surrounding idea is that they are considered interested to share similar interest if can potentially show interests for the same content in the future. Thus, peers collaboratively exchange useful recommendations among themselves.

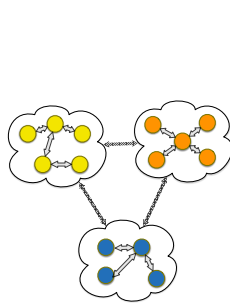


Fig. 1: Interest Communities

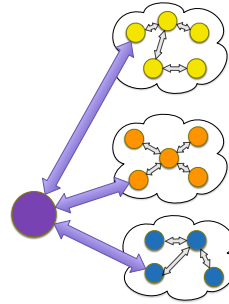


Fig. 2: Interest Overlays

The protocol, to group similar users in communities, adopts a clustering algorithm. As a first step each peer determines, independently, the peers to link

with. These one-to-one connections are established on the basis of an interest-based degree, that is measured amongst the peer it encounters. Every time it becomes aware of a new peer, it can, in turn, learn of the existence of new potential neighbors and possibly communicate with them. Finally, it can also be aware of other, potentially better neighbors. The idea is that this process is stabilized when each node composing the neighborhood of a peer can be considered as the representative of a community of a shared interest. An important side-effect of this vision is that a peer is characterized by multiple interests, with different “entry-points” for each interest. The process is conducted separately for each of the interests of a peer. Consequently, the connections are established and maintained separately for every distinct interest. At the end will be created a set of virtual different overlays, where each peer participates in as many groups is required to cover its interests. The resulting scenario situation is depicted in Fig.2. In order to obtain such organisation, each peer initiate a different stream of messages, one for each of its interests.

### 3.1 Profiles

Profiles of peers need to be modeled according to the users’ interests. A possible approach would be based on recently accessed resources, purchased items, visited pages, etc. Such information, once gathered, has to be considered in a proper way. Basically, it consists in the basis over which the overlay network will be organised. Generally speaking, let  $\mathfrak{S}$  be the set of items belongings to the whole set of profiles of users and let  $\mathfrak{S}_p \subseteq \mathfrak{S}$  be the subset of items belonging to a specific peer  $p$ . We consider that the profile  $\pi$  of  $p$  can be defined as

$$\pi_p = \{(i, C(i), R(i)) | i \in \mathfrak{S}_p\}$$

where  $i$  is an item belongings to the set of  $\mathfrak{S}_p$ ,  $C(i)$  is the content associated with  $i$  and  $R(i)$  is the rating given by  $p$  to  $i$ . The peer  $p$  has also associated a set  $I^p = \{I_1^p, \dots, I_k^p\}$  of interests. Each of the items in  $\pi_p$  may be associated with an interest  $I_j^p$ . We can then represent  $\pi_p$  in the following way:

$$\pi_p = \bigcup_{j=1, \dots, k} \pi_p(I_j^p)$$

where  $\pi_p(I_j^p)$  is the set of items related to the interest  $I_j^p$ . For realising this association, we introduce a function  $\gamma$  that given an item belonging to  $\mathfrak{S}$  decides the interest it should be associated with. More formally:

$$\gamma_p(i) = I_j^p \text{ with } i \in \mathfrak{S}_p$$

It is worth to note that the set  $I^p$  is specific for each distinct peer  $p$ . In fact, we do not assume any globally known labeling, classification or partitioning of the objects in  $\mathfrak{S}$ . Each peer performs its own subdivision of  $\mathfrak{S}_p$  in the interests of  $I_p$ . It can then compare its objects divided per interest with the sets of the other peers it will contact. Given two peers  $p_1$  and  $p_2$ ,  $p_1$  would consider its

local interest  $I_s^{p_1}$  similar to the interest  $I_t^{p_2}$  if it would contain the most similar set of items among the other sets in  $I^{p_2}$  with respect to the items in  $I_s^{p_1}$ . As a consequence of having a solution to describe each user interests coded in the peer profiles, it is important to pay attention on adopting a proper similarity function  $sim : \Pi^2 \rightarrow \mathbb{R}$  to compare profiles, where  $\Pi$  is the set of all possible profiles. This is a key aspect, since this function specify the relationships between peers according to their interests. If each distinct interest is determined by different type of features, different similarity measures could be used to evaluate peers proximities with respect to each interest. Several measures can be adopted to this end. As an example, a typical approach is to use a metric that takes into account the size of each profile, such as the Jaccard similarity, which has proven to be an effective similarity measure [3, 15]. Given two peers  $p_1$  and  $p_2$  and two interests  $I_s^{p_1}$  and  $I_t^{p_2}$ , the similarity can be computed as

$$sim(p_1, p_2) = \frac{|\pi_{p_1}(I_s^{p_1}) \cap \pi_{p_2}(I_t^{p_2})|}{|\pi_{p_1}(I_s^{p_1}) \cup \pi_{p_2}(I_t^{p_2})|}$$

### 3.2 Setup of Interest Communities

One of the base assumptions of our envisioned system is that every peer is able to compute its interest-based distance to any other peer in the network. This measure allows it drive its ability to *group* with other peers that have close-by interests, in order to form the basis for *interests communities*. This process is conducted automagically in a self-organizing and completely decentralized way, using a epidemic communication. Each peer knows a set of other peers, namely its neighbors, and tries periodically to choose new neighbors that are closer to its interest than the previous ones. In our envisioned system, this is simply obtained by discovering new peers from some other peer, then retrieving their profiles. Finally, choosing the  $C$  nearest neighbors in the union of present and potential neighbors. When a peer  $p$  joins the network, it becomes in contact with one or more peers already belonging to in the interest-proximity network overlay. They use the profile similarity function to compute how similar they are. They consider each interest in the  $I$  of  $\pi_p$  separately and they compare it against their own. Furthermore, the peers contacted by  $p$  use the same similarity function to determine which are, among their neighbors, the most similar to  $p$ . Once determined, the join request of  $p$  are routed toward them. All the peers that receive that request will react using the same protocol described above. All the interactions are shown in Algorithms 1 and 2. This approach will lead  $p$  to become aware of the existence of the most similar peers in the network overlay and allow it to connect with them. In doing this process, the involved peers can only use their local knowledge to compare their respective profiles.

Once the process is stabilized,  $p$  can consider its neighbors as the representatives of a personal community of “friends” from which request and to which forward recommendations. Thus, the gossip protocol provide the basis for classical recommender systems in forming the set of similar users. This is done distributively and adaptively and the epidemic protocol ensure a robust and

<p><b>Algorithm 1</b></p> <p>Let <math>CR(P')</math> be a connection request from another peer <math>P'</math></p> <p>Let <math>NewPeers = \emptyset</math></p> <p><b>if</b> <math>Sim(P, P') \geq \min_{P_i \in N(P)} Sim(P, P_i)</math> <b>then</b></p> <p style="padding-left: 20px;">Accept <math>CR(P')</math></p> <p style="padding-left: 20px;"><b>for all</b> <math>P_i \in N(P)</math> <b>do</b></p> <p style="padding-left: 40px;"><b>if</b> <math>Sim(P_i, P') \geq \theta</math> <b>then</b></p> <p style="padding-left: 60px;">add <math>P_i</math> to <math>NewPeers</math></p> <p style="padding-left: 40px;"><b>end if</b></p> <p style="padding-left: 20px;"><b>end for</b></p> <p style="padding-left: 20px;">add <math>P'</math> to <math>N(P)</math></p> <p style="padding-left: 20px;">send <math>NewPeers</math> to <math>P'</math></p> <p><b>else</b></p> <p style="padding-left: 20px;">refuse <math>CR(P')</math></p> <p><b>end if</b></p>	<p><b>Algorithm 2</b></p> <p>Let <math>N(P)</math> be the set of <math>P</math>'s actual neighbors</p> <p><b>for all</b> <math>P_i \in N(P)</math> <b>do</b></p> <p style="padding-left: 20px;">Get from <math>P_i</math> a set <math>NewPeers</math> from its neighborhood</p> <p style="padding-left: 20px;"><b>for all</b> <math>P' \in NewPeers</math> <b>do</b></p> <p style="padding-left: 40px;"><b>if</b> <math>P' \notin N(P)</math> <b>then</b></p> <p style="padding-left: 60px;">connect with <math>P'</math></p> <p style="padding-left: 60px;"><b>if</b> <math>Sim(P, P') \geq \min_{P_j \in N(P)} Sim(P, P_j)</math></p> <p style="padding-left: 80px;"><b>then</b></p> <p style="padding-left: 100px;">add <math>P'</math> to <math>N(P)</math></p> <p style="padding-left: 80px;"><b>end if</b></p> <p style="padding-left: 60px;"><b>end if</b></p> <p style="padding-left: 40px;"><b>end for</b></p> <p style="padding-left: 20px;"><b>end for</b></p>
<p><b>Algorithm 3</b></p> <p>Let <math>N_p(I_j)</math> be the set of <math>P</math>'s neighbors for the interest <math>I_j</math></p> <p>Receive a recommendation request from <math>p' \in N_P(I_j)</math></p> <p><b>for all</b> <math>i \in \pi_{p'}(I_j)</math> <b>do</b></p> <p style="padding-left: 20px;"><b>if</b> <math>Sim(p', i) \geq \theta</math> <b>then</b></p> <p style="padding-left: 40px;">recommend <math>i</math> to <math>p'</math></p> <p style="padding-left: 20px;"><b>end if</b></p> <p style="padding-left: 20px;"><b>end for</b></p>	<p><b>Algorithm 4</b></p> <p>Know about a new item <math>h</math></p> <p>Let <math>I_j</math> be the interest <math>h</math> is related to</p> <p>Let <math>N_P(I_j)</math> be the neighborhood of peers interested in <math>I_j</math></p> <p><b>for all</b> <math>p' \in N_P(I_j)</math> <b>do</b></p> <p style="padding-left: 20px;"><b>if</b> <math>Sim(p', h) \geq \theta</math> <b>then</b></p> <p style="padding-left: 40px;">recommend <math>h</math> to <math>p'</math></p> <p style="padding-left: 20px;"><b>end if</b></p> <p style="padding-left: 20px;"><b>end for</b></p>

Table 1: Active and passive threads and pull and push recommender algorithms

constant maintenance over time. Recommendations can then be requested by  $p$  to its neighborhood and it can forward the newly items it discovered to its neighbors using Algorithms 3 and 4.

## 4 Conclusion

The focus of this paper is on giving a simple recipe for addressing the problem of clustering users in a purely decentralized way to foster information exchange. This is a particularly useful brick for enabling self-emerging and automated creation of communities of nodes representing users, which share common interests. In this paper we sketched the overall architecture of a epidemic-based distributed system exploiting a collaboratively built recommender system. The solution sketched in this work is simpler than most part of the existing solutions. This is inline with our goal: keep the solution as simple as possible but still providing a solution that exploit collaborative filtering is able to provide recommendations that are tailored and offer an acceptable degree of serendipity.

## References

1. Emanuele Carlini, Laura Ricci, and Massimo Coppola. Reducing server load in mmog via p2p gossip. In *Proceedings of the 11th Annual Workshop on Network and Systems Support for Games*, page 11. IEEE Press, 2012.

2. Emanuele Carlini, Massimo Coppola, and Laura Ricci. Evaluating compass routing based aoi-cast by mogs mobility models. In *Proceedings of the 4th International ICST Conference on Simulation Tools and Techniques*, pages 328–335. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2011.
3. Pierre Fraigniaud, Philippe Gauron, and Matthieu Latapy. Combining the use of clustering and scale-free nature of user exchanges into a simple and efficient p2p system. In *Proc. of EuroPar’05*, 2005.
4. Sidath B. Handurukande, Anne-Marie Kermarrec, Fabrice Le Fessant, Laurent Massoulié, and Simon Patarin. Peer sharing behaviour in the edonkey network, and implications for the design of server-less file sharing systems. In *Proc. of Eurosys’06*, Leuven, Belgium, apr 2006.
5. Emanuele Carlini, Massimo Coppola, Patrizio Dazzi, Domenico Laforenza, Susanna Martinelli, and Laura Ricci. Service and resource discovery supports over p2p overlays. In *Ultra Modern Telecommunications & Workshops, 2009. ICUMT’09. International Conference on*, pages 1–8. IEEE, 2009.
6. Ranieri Baraglia, Patrizio Dazzi, Matteo Mordacchini, and Laura Ricci. A peer-to-peer recommender system for self-emerging user communities based on gossip overlays. *Journal of Computer and System Sciences*, 79(2):291–308, 2013.
7. Patrizio Dazzi, Pascal Felber, Lorenzo Leonini, Matteo Mordacchini, Raffaele Perego, Martin Rajman, and Étienne Rivière. Peer-to-peer clustering of web-browsing users. *Proc. LSDS-IR*, pages 71–78, 2009.
8. Ranieri Baraglia, Patrizio Dazzi, Matteo Mordacchini, Laura Ricci, and Luca Alessi. Group: A gossip based building community protocol. In *Smart Spaces and Next Generation Wired/Wireless Networking*, pages 496–507. Springer Berlin Heidelberg, 2011.
9. Matteo Mordacchini, Patrizio Dazzi, Gabriele Tolomei, Ranieri Baraglia, Fabrizio Silvestri, and Salvatore Orlando. Challenges in designing an interest-based distributed aggregation of users in p2p systems. In *Ultra Modern Telecommunications & Workshops, 2009. ICUMT’09. International Conference on*, pages 1–8. IEEE, 2009.
10. Patrizio Dazzi, Matteo Mordacchini, and Fabio Baglini. Experiences with complex user profiles for approximate p2p community matching. In *Computer and Information Technology (CIT), 2011 IEEE 11th International Conference on*, pages 53–58. IEEE, 2011.
11. Emanuele Carlini, Patrizio Dazzi, Matteo Mordacchini, and Laura Ricci. Toward community-driven interest management for distributed virtual environment. In *Euro-Par 2013: Parallel Processing Workshops*, pages 363–373. Springer Berlin Heidelberg, 2014.
12. Matteo Mordacchini, Patrizio Dazzi, Ranieri Baraglia, and Laura Ricci. Towards group protocol formalization. In *Peer-to-Peer Computing (P2P), 2013 IEEE Thirteenth International Conference on*, pages 1–2. IEEE, 2013.
13. Moreno Marzolla, Matteo Mordacchini, and Salvatore Orlando. A p2p resource discovery system based on a forest of trees. In *Database and Expert Systems Applications, 2006. DEXA’06. 17th International Workshop on*, pages 261–265. IEEE, 2006.
14. Claudio Gennaro, Matteo Mordacchini, Salvatore Orlando, and Fausto Rabitti. Mroute: A peer-to-peer routing index for similarity search in metric spaces. *Proceedings of the 5th International Workshop on Databases, Information Systems and Peer-to-Peer Computing (DBISP2P 2007)*, pages 1–12, 2007.

15. Jay A. Patel, Etienne Rivière, Indranil Gupta, and Anne-Marie Kermarrec. Rappel: Exploiting interest and network locality to improve fairness in publish-subscribe systems. *Computer Networks*, 2009. In Press.
16. Bin Tan, Xuehua Shen, and ChengXiang Zhai. Mining long-term search history to improve search accuracy. In *Proc. of SIGKDD'06*, pages 718–723, Philadelphia, PA, USA, 2006.
17. Jaime Teevan, Susan T. Dumais, and Eric Horvitz. Personalizing search via automated analysis of interests and activities. In *Proc. of SIGIR-IR'05*, pages 449–456, Salvador, Brazil, 2005.
18. David Cheng, Ravi Kannan, Santosh Vempala, and Grant Wang. A divide-and-merge methodology for clustering. *ACM Trans. Database Syst.*, 31(4):1499–1525, 2006.
19. Fatih Gelgi and Hasan Davulcu Srinivas Vadrevu. Term ranking for clustering web search results. In *Proc. of WebCD 2007*, Beijing, China, jun 2007.
20. Karl Aberer, Philippe Cudre-Mauroux, Manfred Hauswirth, and Tim Van Pelt. Gridvine: Building internet-scale semantic overlay networks. In *In Proc. of ISWC 04*, pages 107–121, 2004.
21. Arturo Crespo and Hector Garcia-Molina. Semantic overlay networks for p2p systems. Technical report, cs department, Stanford University, 2002.
22. Josiane Xavier Parreira, Sebastian Michel, and Gerhard Weikum. p2pdating: Real life inspired semantic overlay networks for web search. *Inf. Process. Manage.*, 43(3):643–664, 2007.
23. J. A. Pouwelse, P. Garbacki, J. Wang, A. Bakker, J. Yang, A. Iosup, D. H. J. Epema, M. Reinders, M. R. van Steen, and H. J. Sips. Tribler: a social-based peer-to-peer system. *Concurrency and Computation: Practice and Experience*, 20(2):127–138, 2008.